



# Vehicle de guiat automàtic (AGV)

Memòria del projecte  
d'Enginyeria Tècnica en  
Informàtica de Sistemes  
realitzat per

*Enric Marco Nin*

i dirigit per

*Lluís Ribas-Xirgo*

**Escola d'Enginyeria**

*Sabadell, Setembre de 2012*



El sotasignat, **Lluís Ribas-Xirgo**,  
professor de l'Escola d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball al que correspon la present memòria  
ha estat realitzat sota la seva direcció per

**Enric Marco Nin**

I per a que consti firma la present.

Sabadell, **setembre** de **2012**

-----  
Signat: **Lluís Ribas-Xirgo**



**FULL DE RESUM – PROJECTE FI DE CARRERA DE L'ESCOLA D'ENGINYERIA**

<b>Títol del projecte:</b> Vehicle de guiat automàtic (AGV)	
<b>Autor:</b> Enric Marco Nin	<b>Data:</b> Setembre 2012
<b>Tutor:</b> Lluís Ribas-Xirgo	
<b>Titulació:</b> Enginyeria Tècnica en Informàtica de Sistemes	
<b>Paraules clau</b> <ul style="list-style-type: none"> <li>• <u>Català</u>: AGV, vehicle, guiat, automàtic, simulació, anàlisis clínic, robot, decisió, Basic Stamp.</li> <li>• <u>Castellà</u>: AGV, vehículo, guiado, automático, análisis clínico, robot, decisión, Basic Stamp.</li> <li>• <u>Anglès</u>: AGV, automatic, guided, vehicle, simulation, clinical analysis, robot, decision, Basic Stamp.</li> </ul>	
<b>Resum del projecte</b> (extensió màxima 100 paraules) <ul style="list-style-type: none"> <li>• <u>Català</u>: Aquest projecte documenta la simulació d'un AGV (vehicle de guiat automàtic).</li> </ul> <p>Mitjançant un robot educatiu programat en Basic Stamp és simula un entorn d'anàlisi clínic a on el robot gestiona de forma automàtica les estacions d'anàlisi d'una hipotètica mostra de sang.</p> <li>• <u>Castellà</u>: Este proyecto documenta la simulación de un AGV (vehículo de guiado automático ).</li> <p>Utilizando un robot educativo programado en Basic Stamp se simula un entorno análisis clínico donde el robot gestiona de forma automática las estaciones de análisis de una hipotética muestra de sangre.</p> <li>• <u>Anglès</u>: This project documents the simulation of an AGV (automatic guided vehicle).</li> <p>Using an educational robot programmed in Basic Stamp a clinical analysis environment is simulated where the robot automatically manages the analysis stations of a hypothetical blood sample.</p>	



# Índex

<b>1. Introducció .....</b>	<b>3</b>
1.1 Context .....	3
1.2 Motivació .....	4
1.3 Objectius .....	5
1.4 Tasques .....	5
1.5 Planificació .....	6
1.6 Anàlisi de viabilitat .....	10
1.6.1 Estimació de costos .....	10
1.6.2 Anàlisi de riscos .....	13
 <b>2. Especificacions .....</b>	 <b>15</b>
2.1 Entorn de treball .....	15
2.2 Model de l'AGV .....	16
2.3 Àrea de treball .....	17
2.4 Problemes .....	19
 <b>3. Vehicle robòtic .....</b>	 <b>20</b>
3.1 Control dels AGV .....	20
3.1.1 Unió .....	20
3.1.2 Bifurcació .....	20
3.1.3 Parada de programació .....	21
3.1.4 Parada per obstacle .....	21
3.1.5 Recirculació .....	21
3.1.6 Entorn de programació .....	21
3.2 Programa de control .....	22
3.2.1 Subrutina: Enviar motors .....	23

3.2.2	Subrutina: Parada per obstacle .....	25
3.2.3	Subrutina: Bifurcació .....	27
3.2.4	Subrutina: Unió .....	31
3.2.5	Subrutina: Parada de programació .....	33
3.2.6	Subrutina: Recirculació .....	35
3.2.7	Subrutina: Seguir línia .....	37
3.2.8	Rutina Principal .....	39
<b>4.</b>	<b>Proves i errors.....</b>	<b>45</b>
4.1	Proves i casos de test .....	45
4.2	Errors durant el desenvolupament .....	51
<b>5.</b>	<b>Conclusions .....</b>	<b>52</b>
5.1	Compliment d'objectius .....	52
5.2	Desviacions .....	52
5.3	Línies d'ampliació .....	53
	<b>Referències .....</b>	<b>54</b>
	<b>Annexos .....</b>	<b>55</b>



# Capítol 1. Introducció

## 1.1. Context

Un AGV o vehicle de guiat automàtic, com la descripció del nom indica, el podríem definir com a un vehicle o robot que es mou de forma automàtica amb uns objectius/destins fixats, és a dir, de forma que ningú el comandi, i que està destinat a realitzar una tasca concreta en un entorn prefixat.

Els AGV són vehicles que varien en forma i mida en funció de la feina que són programats a fer. Els podem trobar en funcions molt diverses, però normalment els solem trobar en tasques industrials de manufactura, o en transport en magatzems, com ara en ports marítims a on normalment s'encarreguen del transport intern dels contenidors dels vaixells. Una altre tasca que desenvolupen pot ser la de transport de peces en un procés de manufactura a on l'AGV recull una peça tractada per una màquina o operari, i la transporta fins al següent punt dins de la cadena de manufactura.

Per a poder realitzar les tasques a les quals són destinats, és necessari que l'AGV conegui el seu entorn de treball, és a dir, per a on s'ha de moure. Normalment per a que això sigui possible és fan servir vies/línies o balises. En el cas de les vies, l'AGV té el seu recorregut estrictament marcat sabent així a on s'ha d'aturar per a que pugui carregar un peça manufacturada, un contenidor portuari... i poder seguir fins al següent destí. En el cas de les balises, l'objectiu és similar però la manera de moure's no és tant marcada ja que pot admetre canvis en la seva trajectòria d'una balisa a un altra, permetent així que tant operaris, com altres AGVs puguin passar per la possible trajectòria de l'AGV i aquest sigui capaç de rectificar el seu moviment, així com aturar-se en cas de trobar un obstacle.

Per limitar l'abast de la temàtica, el projecte es centrarà en el possible ús de mini-AGV en els laboratoris d'anàlisis clínics [1] que, en els darrers anys, s'han convertit en plantes altament complexes.

Els laboratoris d'anàlisis són, de fet, plantes de manufactura on, a partir de mostres de sang, es produeixen anàlisis que contenen determinacions (dades de les mesures efectuades) de les mostres que es processen.

## 1.2. Motivació

Les millores són significatives en l'ús d'AGVs per a dur a terme segons quines tasques ja que suposen un augment significatiu en la producció o logística implícites en el transport, al ser capaços d'estar operant durant llargs intervals de temps (24 hores) sense necessitat de que ningú els condueixi.

En el cas en que en un procés de manufactura es trobi amb un error o retràs en una de les seves màquines, o en el cas del transport, que quan s'arribi al punt de descarrega no es pugui descarregar la mercaderia, l'AGV és capaç d'aturar-se o bé anar cap a un altre punt a on acabar la seva tasca.

D'altra banda, si el vehicle ho permet en les seves característiques, pot ser aprofitat per a realitzar tasques diverses si és necessari.

Els AGV per tant, aporten flexibilitat i tolerància a fallades. El desenvolupament de les plantes s'ha de verificar prèviament, per a això, convé disposar d'algun sistema de simulació, bé sigui per software o amb prototips.

En qualsevol cas, els sistemes d'AGV impliquen el trànsit de vehicles en vials més o menys definits de l'àrea de treball, cosa que significa que s'han de resoldre els problemes inherents al mateix trànsit de vehicles: Hi ha d'haver unes normes de circulació conegudes per tots els AGV i també una regulació del trànsit, bé sigui global, amb un sistema de gestió unificat que resol els possibles conflictes a les travesseres de circulació o, de forma més o menys local, de manera que siguin els mateixos AGV que determinen com resoldre els conflictes de circulació que es van trobant. Se suposa que tots els agents del sistema (tant la part de gestió com els mateixos AGV) coneixen el mapa de l'àrea de treball.

## 1.3. Objectius

Objectiu principal:

- Emular un AGV en un laboratori d'anàlisi clínic amb un mini robot educatiu seguidor de línies.

Objectius secundaris:

- Construcció del mini robot
- Simular una planta de laboratori d'anàlisi clínic per a on operará el mini robot.
- Comunicar-se amb el mini robot per a indicar les tasques a realitzar.

## 1.4. Tasques

- Establir un mapa per a la planta del laboratori, a on hi hagi els recorreguts possibles i les estacions d'anàlisi a on ha d'anar el mini robot.
- Marcar el recorregut que ha de fer el robot amb les línies que ha de seguir.
- Trobar una metodologia per a decidir sobre les línies quina decisió farà el mini robot.
- Construir el mini robot.
- Implementar el programa que controli el moviment del mini robot, en les línies del recorregut.
- Codificar les estacions a on el robot s'haurà d'aturar així com els possibles canvis de direccions en les línies.
- Implementar el programa que permeti comunicar-se amb el mini robot per a marcar l'itinerari de les estacions d'aturada.

## 1.5. Planificació

Aquest punt recull la planificació del projecte, és a dir, el conjunt d'activitats que permeten desenvolupar, executar i controlar el projecte, així com el calendari temporal del projecte amb les seves tasques associades. És trobarà una breu descripció de les tasques així com de la planificació en hores i recursos que aquestes requereixen.

Fases i activitats del projecte:

Fases	Descripció
Iniciació	Fase d'iniciació. Inclou les activitats de definició del projecte, assignació i matriculació.
Planificació	Inclou l'Estudi de Viabilitat i el Pla del Projecte.
Anàlisi	Anàlisi de requeriments del projecte. Arquitectura del sistema.
Disseny	Inclou el disseny de l'entorn de treball del robot i el programa que el comandi.
Desenvolupament	Fase de desenvolupament del programa del robot així com de l'entorn de treball.
Test i proves	Fase de prova del sistema. Inclou tests del sobre l'entorn de treball.
Generació de documents	Fase de documentació del projecte. Inclou la memòria del projecte.
Tancament del projecte	Fase de tancament. El director del projecte signa l'acceptació i tancament del projecte.
Defensa del projecte	Defensa del projecte davant la comissió.

## *Fites importants en el projecte:*

Nom	Descripció	Data
Iniciació	Matriculació	06/10/11
Estudi de Viabilitat i Pla del Projecte	Aprovació	14/12/11
Anàlisi	Aprovació	01/03/12
Disseny i desenvolupament	Aprovació	23/06/12
Tancament	Acceptació	19/09/12
Defensa	Avaluació	27/09/12

Totes les fases es desenvolupen utilitzant un model lineal, per tant, cada fase no es comença fins que no s'ha completat la fase anterior.

En la fase de desenvolupament es preveu un model àgil de tal manera que les fases de disseny, desenvolupament i de test segueixin un model iteratiu, així serà més senzill detectar-hi errors i corregir-los.

La fase de generació de documents es preveu realitzar-la a mesura que el projecte avança. D'aquesta manera no es deixa cap acte, fita, ni hora de feina sense documentar, i per altre banda, la memòria es confecciona progressivament ajudant a la fase d'elaboració de documents, ja que llavors es tractarà d'unificar tota la documentació.

Per tant, disposem ja d'unes etapes que ja han estat executades abans de realitzar la planificació del projecte, ja que la planificació no ha començat fins que els objectius han quedat fixats (punt 1.3).

Altres etapes ja executades són la cerca d'informació prèvia sobre que es un AGV (punt 1.1) o la motivació que aporta a la realització d'aquest projecte (punt 1.2).

### – Calendari general del projecte:

- Data començament: 06 d'octubre de 2011
- Data finalització: 19 de setembre de 2012
- Eines de planificació i control: Microsoft Project 2010

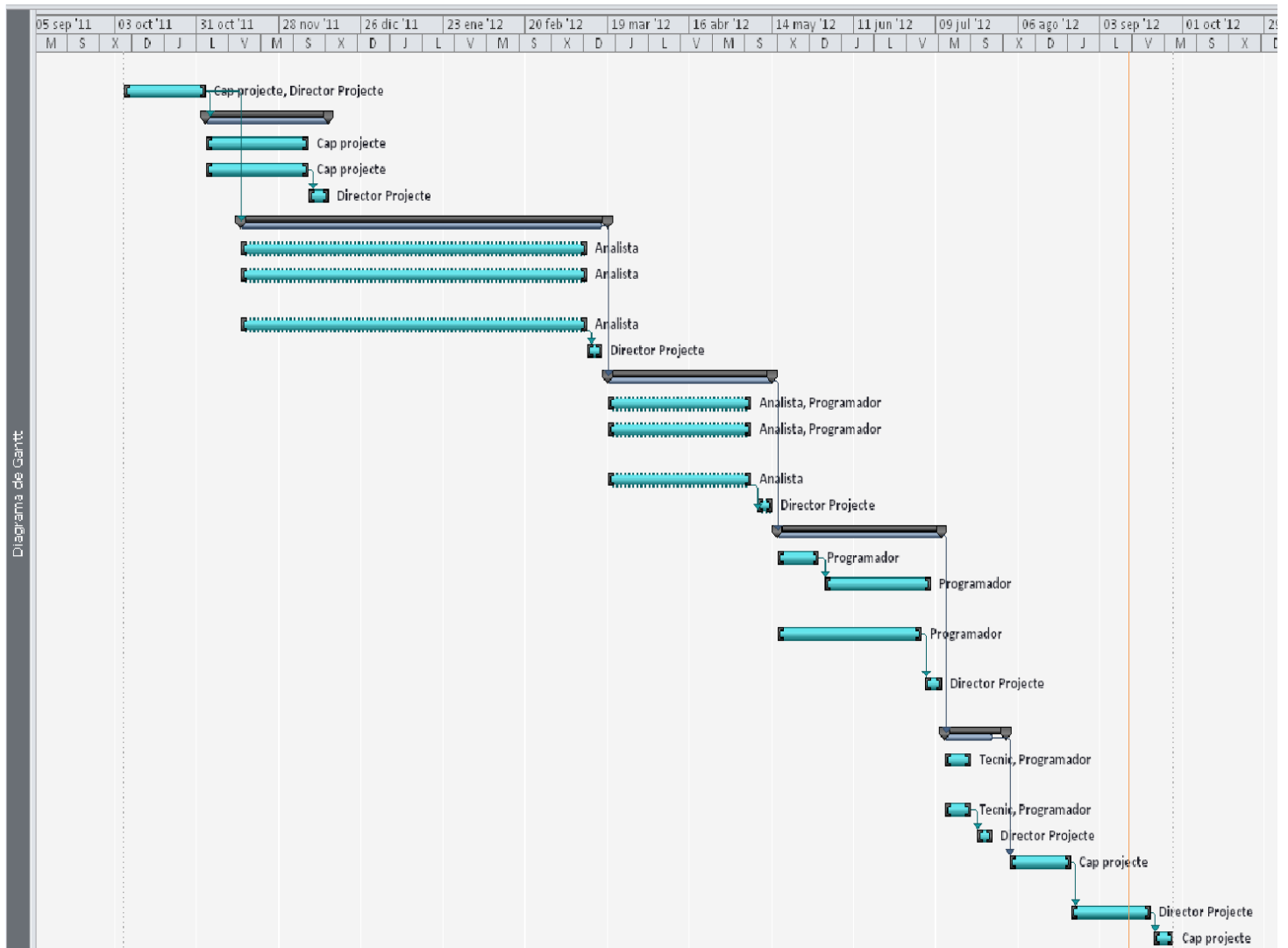
- Quadre de tasques del projecte:

Elaborat amb Microsoft Project 2010:

	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1							
2		Inici del projecte	20 días	jue 06/10/11	mié 02/11/11		Cap projecte, Director Projecte
3		Planificació	30 días	jue 03/11/11	mié 14/12/11	2	
4		Estudi de viabilitat	25 días	jue 03/11/11	mié 07/12/11		Cap projecte
5		Planificació	25 días	jue 03/11/11	mié 07/12/11		Cap projecte
6		Aprovació de la planificació	5 días	jue 08/12/11	mié 14/12/11	5	Director Projecte
7		Anàlisi	90 días	mar 15/11/11	dom 18/03/12	2	
8		Anàlisi d'especificacions	85 días	mar 15/11/11	dom 11/03/12		Analista
9		Anàlisi de l'arquitectura del sistema	85 días	mar 15/11/11	dom 11/03/12		Analista
10		Documentació de l'anàlisi	85 días	mar 15/11/11	dom 11/03/12		Analista
11		Aprovació de l'Anàlisi	5 días	lun 12/03/12	vie 16/03/12	10	Director Projecte
12		Disseny	41 días	lun 19/03/12	dom 13/05/12	7	
13		Disseny de l'entorn de treball	36 días	lun 19/03/12	dom 06/05/12		Analista, Programador
14		Disseny del pseudocodi del programa del robot	36 días	lun 19/03/12	dom 06/05/12		Analista, Programador
15		Documentació del disseny	36 días	lun 19/03/12	dom 06/05/12		Analista
16		Aprovació del disseny	4 días	mié 09/05/12	dom 13/05/12	15	Director Projecte
17		Desenvolupament	40 días	mié 16/05/12	mar 10/07/12	12	
18		Construcció de l'entorn de treball	10 días	mié 16/05/12	mar 29/05/12		Programador
19		Codificació del programa de control del robot	26 días	vie 01/06/12	vie 06/07/12	18	Programador
20		Documentació del desenvolupament	35 días	mié 16/05/12	mar 03/07/12		Programador
21		Aprovació del desenvolupament	4 días	jue 05/07/12	mar 10/07/12	20	Director Projecte
22		Test i proves	15 días	jue 12/07/12	mié 01/08/12	17	
23		Proves d'estrès (incidències i riscos)	7 días	jue 12/07/12	vie 20/07/12		Tecnic, Programador
24		Documentació dels test	7 días	jue 12/07/12	vie 20/07/12		Tecnic, Programador
25		Aprovació dels test	5 días	lun 23/07/12	vie 27/07/12	24	Director Projecte
26		Generació de documents (memòria)	15 días	vie 03/08/12	jue 23/08/12	22	Cap projecte
27		Tancament del Projecte	19 días	vie 24/08/12	mié 19/09/12	26	Director Projecte
28		Defensa del Projecte	5 días	vie 21/09/12	jue 27/09/12	27	Cap projecte

- Calendari temporal:

Diagrama de Gantt elaborat amb Microsoft Project 2010:



## 1.6. Anàlisi de viabilitat

La viabilitat del projecte ve implícita degut a que es tracta de un model a escala d'un tipus de projecte significativament molt més gran.

Al estar tractant amb un projecte que emula el comportament d'un projecte major, el cost del qual pot assolir els milions d'euros, realitzar el model a escala es quelcom útil i necessari en el suposat de voler acabar implementant el model real.

Aquest avantatge es degut a que la adaptació del model a escala es relativament senzilla, ja que, el que canvia és l'entorn i la maquinària, però en el cas del funcionament del model, es pràcticament el mateix.

Per tant, en el cas de voler implementar un sistema real basat en AGV, es extremadament aconsellable el realitzar aquests tipus de projectes a escala per a veure si el sistema real pot assolir bons resultats sense haver de fer la inversió milionària.

L'anàlisi de viabilitat d'aquest projecte (el projecte a escala), es detalla a continuació:

### 1.6.1. Estimació de costos

- Recursos humans i materials :

<b><i>Recursos humans[2]</i></b>	<b><i>Valoració</i></b>
Director de projecte	48 €/h
Cap de projecte	30 €/h
Analista	30 €/h
Programador	25 €/h
Tècnic de proves	19 €/h



<b>Recursos materials</b>	<b>Cost</b>
Boe-bot kit (mini-robot + sensors) [3]	120,00 €
Cinta Americana [4]	4,98 €
Carregador de piles + piles [4]	24,07 €
<b>Total</b>	<b>164,32 €</b>

- Estimació de cost del personal:

<b>Recursos humans</b>	<b>Temps</b>	<b>Cost</b>
<b>Director de projecte</b>	30 h	1.440,00 €
<b>Cap de projecte</b>	45 h	1.350,00 €
<b>Analista</b>	65 h	1.950,00 €
<b>Programador</b>	75 h	1.875,00 €
<b>Tècnic</b>	15 h	285,00 €

**TOTAL : 6900 €**

- Estimació de costos dels recursos:

Suposem que per al sistema d'emulació són necessaris 5 mini-robots:

5 mini-robots ..... 600 €  
 Material per l'entorn d'emulació ..... 29,02 €

**TOTAL : 629,02 €**

– Estimació de cost total del projecte:

Costos de personal ..... 6900 €

Costos de recursos ..... 629,02 €

**TOTAL : 7529,02 €**

## 1.6.2. Anàlisi de riscos

A continuació es detalla una llista amb els possibles riscos que poden afectar al projecte, així com les conseqüències que porten sobre la planificació del projecte.

– Llista de riscos:

**R1. Planificació temporal optimista:** Afecta sobre el pla de projecte. El projecte no s'acaba en la data prevista, i per tant augmenten els recursos.

**R2. Manca alguna tasca necessària:** Afecta sobre el pla de projecte. No es compleixen els objectius del projecte.

**R3. Pressupost poc ajustat:** Afecta sobre el pla de projecte i la viabilitat econòmica. Augmenten els costos dels recursos.

**R4. Canvi de requisits de l'entorn d'emulació:** Afecta sobre el pla de projecte. Es produeix un endarreriment en les tasques d'anàlisi i disseny.

**R5. Canvi del mini-robot:** Afecta sobre el pla de projecte i l'anàlisi de viabilitat. Es produeix un endarreriment en les tasques d'anàlisi i disseny i una variació en els costos de material.

**R6. No es fa correctament la fase de test:** Afecta a la fase de desenvolupament. Manca de qualitat en els resultats finals, deficiències en el funcionament, pèrdua econòmica.

– Catalogació de riscos:

	Probabilitat	Impacte
<b>R1</b>	Alta	Crític
<b>R2</b>	Alta	Crític
<b>R3</b>	Alta	Normal
<b>R4</b>	Mitja	Normal
<b>R5</b>	Baixa	Normal
<b>R6</b>	Mitja	Crític

- Pla de contingència:

	<b>Solució que cal adoptar</b>
<b>R1</b>	Ajornar alguna funcionalitat, ampliar recursos humans.
<b>R2</b>	Revisar el Pla de Projecte, modificar la planificació.
<b>R3</b>	Afrontar possibles pèrdues, augmentar pressupost previst.
<b>R4</b>	Augmentar recursos humans en les fases d'anàlisis i disseny.
<b>R5</b>	Augment de recursos humans per adaptar-se al nou robot, ajustar el pressupost previst.
<b>R6</b>	Ajornar alguna funcionalitat, ampliar recursos humans.

# Capítol 2. Especificacions

## 2.1. *Entorn de treball*

Els laboratoris d'anàlisi clínic en els hospitals treballen amb un flux abundant de mostres, tant de sang, com teixits... que han de ser analitzades per maquinària mèdica concreta per a cada mostra i a on cada instrument mèdic dona un resultat en funció del seu anàlisi.

Es donen casos en que mostres del mateix tipus han de ser analitzades per maquinària diferent fent així necessari una medi de transport efectiu entre maquinària.

Aquest laboratoris operen amb cintes de transport similars a les d'una planta embotelladora, podent dir així que es pot entendre el procés d'anàlisi com a un procés de "manufactura d'anàlisi de mostres".

Donats els requeriments d'aquest tipus de laboratoris, ens trobem davant d'un sistema a on el transport entre estacions mèdiques d'anàlisi podrien dur-lo a terme mini-AGVs.

Aquest AGVs per tant, desenvoluparien la funció de transport de mostres clíniques entre les estacions d'anàlisi del laboratori, i en casos en que una mostra hagués de ser repetida o ens trobéssim davant una mostra errònia no caldria aturar tot el procés d'anàlisi com pot passar amb les cintes de transport.

D'altra banda si algunes mostres tenen patrons d'anàlisi, és a dir, que sempre van d'unes estacions específiques a unes altres, amb l'ús d'AGVs es podria donar flexibilitat al sistema, ja que les mostres concretes anirien a les estacions concretes sense necessitat de passar per tota la cinta de transport. També en cas d'error o fallada d'un mini-AGV, serien només aquelles les mostres afectades per la falla, podent operar la resta d'AGVs de forma normal.

## 2.2. Model de l'AGV

El mini robot que és mourà per l'entorn de treball emulant la situació del laboratori d'anàlisi clínic és el robot educatiu Boe-Bot de Parallax (figura 2.1). Aquest robot disposa d'un microcontrolador per a poder executar les instruccions que s'hagin programat per al seu degut comportament, sent l'encarregat dels motors del robot i per tant del seu moviment i també servirà per a interpretar les dades dels sensors.

Altrament disposa de sensors que són necessaris per a desenvolupar la tasca de moure's per l'entorn de treball adequadament.

Sensors del mini robot Boe-Bot:

- Dos sensors d'IR (i un d'emissor) mirant cap al terra per al control de seguiment de línia.
- Un sensor d'IR addicional que mira al terra per a la detecció de marques laterals.
- Un sensor d'ultrasons per detectar distàncies a objectes en la trajectòria.
- Un mòdul de bluetooth per a la comunicació amb el mini-robot.

D'aquesta manera, el mini robot, amb els sensors esmentats i el programa de control per al microcontrolador, pot seguir línies, detectar-hi marques laterals, calcular distàncies a obstacles i rebre informació de l'itinerari a seguir per bluetooth.

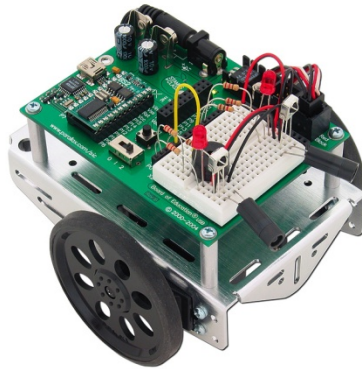


Figura 2.1. Boe-Bot de Parallax [3]

## 2.3. Àrea de treball

Per realitzar el circuit que emuli l'entorn de treball, el robot seguirà una línia construïda amb cinta de vinil i per a poder facilitar la lectura de dades dels sensor per aconseguir que el recorregut del robot circuli per la línia, aquestes no tindran revolts d'angles superiors als  $45^\circ$  afavorint així el traçat del robot.

Durant el traçat del robot per la línia principal es trobarà marques laterals a la banda dreta del circuit perquè el robot pugui determinar en quin punt del mapa de l'àrea de treball es troba, sabent així si s'ha d'aturar ja que ha arribat a destinació, si se li ordenen nous objectius o be per a gestionar les bifurcacions i unions que donen camí a les balises o avançaments.

Les bifurcacions seran línies sortints a l'esquerra de la via principal, amb una de les marques laterals situada prèviament.

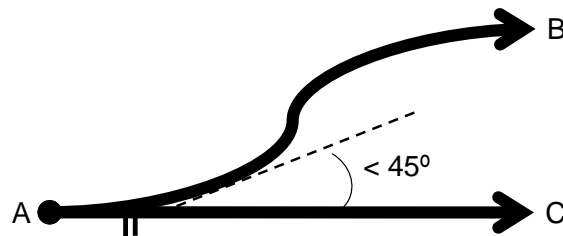


Figura 2.1. Bifurcació

Les unions són punts de confluència de les línies, que també aniran amb marca prèvia i, per facilitar el control, amb una marca posterior que permeti, al vehicle sortint, avisar que la unió ha quedat desocupada.

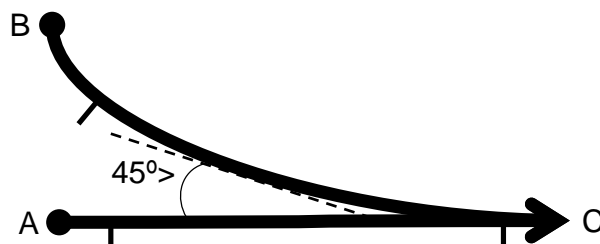


Figura 2.2. Unió

### Mapa de l'àrea de treball

Amb les restriccions esmentades anteriorment l'àrea de treball queda definida com a un circuit tancat amb un únic sentit de circulació, com es mostra a la figura següent, ja que només es disposa d'una via principal.

El circuit disposa d'un bucle exterior, que comença i acaba a la parada 1, i un bucle interior, per permetre la recirculació dels AGV amb destinacions pendents. El traçat disposa de dues zones d'avançament que permetran operar a varis robots amb diferents velocitats.

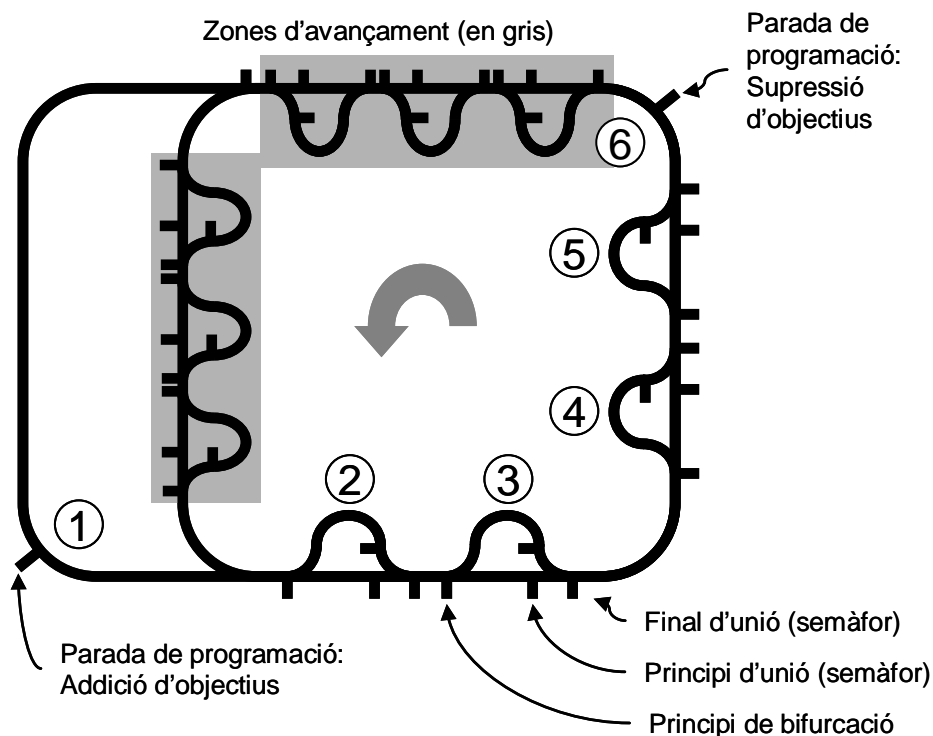


Figura 2.3. Mapa de carreteres de la planta de treball dels AGV

L'operació en una planta com l'anterior comença amb l'AGV que arriba al punt de parada número 1, on rep les destinacions objectiu. Les destinacions es donaran amb una seqüència de dígit a partir del hyperterminal del ordinador, el format de la qual serà:

número destinació – número destinació alternativa ... – número destinació alternativa –  
 ...  
 número destinació – número destinació alternativa ... – número destinació alternativa –  
 9

És a dir, cada objectiu es marca amb el número d'estació que correspon a mode de seqüència. El final de la seqüència és marca amb un 9.



Per exemple: 2–3–5–9 indica a l'AGV que ha d'anar a la parada 2, a la 3 i també a la 5.

A la parada 6 es rebrà una nova seqüència d'objectius, per tant, les destinacions a ser suprimides de la llista d'objectius pendents s'aniran actualitzant en funció de que es realitzin els objectius. En aquest cas, altre vegada serà una sèrie de números acabada en 9. En el cas de que quedin objectius pendents, en aquesta parada es podran afegir nous objectius.

Per exemple, en el cas anterior, si, en arribar a 6, l'AGV rep la seqüència 2–5–9, s'afegiran els objectius 2 i 5 als que ja tingués.

En absència d'objectius, un AGV retornarà a la parada número 1 i no trencarà pel carril de recirculació.

## 2.4. Problemes

Els AGV coneixen el mapa de l'àrea de treball i, consegüentment, com arribar a les destinacions objectius que se'ls fixa. Ara bé, això no treu que hagin de resoldre els conflictes de circulació en les situacions següents:

En una bifurcació, han de decidir si la prenen o no. La decisió depèn de la ruta que tenen planificada. En el cas que es planteja, les bifurcacions permeten arribar a les diferents destinacions o continuar el camí cap a d'altres destinacions. Si una bifurcació porta a una destinació objectiu del vehicle, aleshores, l'única opció és prendre-la. Malauradament, el cas que es planteja permet tenir diverses destinacions per a un mateix objectiu. En aquest cas, la decisió de seguir una bifurcació o continuar en el camí principal fins a una altra destinació amb la que assolir el mateix objectiu es converteix en una decisió que es pot prendre localment o globalment.

En les unions, el conflicte es pot produir quan dos vehicles hi arriben al mateix temps: cal decidir quin passa i quin cedeix el pas. Altre cop, la decisió la poden prendre localment, entre els vehicles que han entrat en conflicte, o globalment, de manera que és el sistema de gestió qui controla la unió amb semàfors.

# Capítol 3. Vehicle robòtic

## 3.1. Control dels AGV

El control dels robots mòbils segueix el cicle següent:

1. Lectura de sensors (de distància a obstacle, de línia, de marca i de comunicació)
2. Deliberació de l'acció a fer segons estat intern i dades dels sensors
3. Actuació (sobre els motors)

En el cas dels Boebot, és convenient que aquest cicle sigui inferior als 20mS per aconseguir un moviment continu, ja que els servomotors requereixen un pols de control cada 20mS. En els casos que el vehicle estigui en una parada (origen o destinació), el cicle pot ser més gran, ja que el que importa és la comunicació, amb els comandaments a distància per rebre ordres externes.

La deliberació de l'acció a fer ha de tenir en compte la localització del vehicle i l'acció que estava fent anteriorment. En general, és una mena d'alternatives en cascada començant per aquelles opcions més prioritàries. Per exemple, si hi ha un obstacle al davant (sensor) i el vehicle es movia endavant (estat), cal aturar-se (acció) sense canviar d'estat.

### 3.1.1. Unió

Davant de la unió de dues vies, cal l'ús de marques que es faran servir com a semàfors. En aquest cas, si un robot arriba a una marca de semàfor d'unió, i aquest semàfor està en "verd" indicarà que la unió està disponible, per tant no hi haurà un altre robot a la via i es podrà seguir el camí, en el moment en que es fa servir la unió, el semàfor es posarà en "vermell" indicant així als possibles robots que s'apropin a la unió, que s'aturin al trobar la marca. Una vegada acabada la unió, el robot trobarà la marca de fi d'unió i alliberarà la via posant el semàfor en "vermell".

### 3.1.2. Bifurcació

Per a la presa de decisió del robot davant d'una bifurcació, es prioritzarà el fet de que el robot tingui com objectiu la balisa a la qual es pot accedir. Per tant, si en detectar una marca lateral que indica un objectiu, i aquest, està en les ordres del robot, es desviarà cap a aquesta bifurcació. En el cas contrari, el robot seguirà per la línia principal.

### 3.1.3. Parada de programació

En els punts en que calgui una parada de programació (punts 1 i 6), el robot s'aturarà esperant les ordres enviades des de l'ordinador a través de bluetooth, i a la que es detecti un final de seqüència d'ordres (número 0), el robot continuarà el seu camí.

### 3.1.4. Parada per obstacle

Sempre que un robot detecti un obstacle, s'aturarà fins a que aquest obstacle estigui fora de l'abast del sensor. Una vegada no es detecti l'obstacle, el robot continuarà. D'aquesta manera si el robot detecta un altre robot (obstacle) degut a que aquest es troba en una parada de programació, s'aturarà fins que el robot obstacle hagi acabat la seva parada.

### 3.1.5. Recirculació

En el moment en que el robot trobi la marca de recirculació, que serà un cas especial de bifurcació, el robot comprovarà si disposa d'objectius pendants, si és el cas, agafarà la bifurcació de recirculació. Si el robot ja no té objectius, seguirà endavant fins arribar a la parada programada número 1.

### 3.1.6. Entorn de programació

Per a la programació del microcontrolador que controlarà els paràmetres dels sensors així com el moviment del robot, és farà servir el software de Parallax per al robot Boe-Bot denominat "BASIC Stamp" [5] i el llenguatge per a programar el microcontrolador i per tant el robot es el PBASIC [5].

El software per a la programació del Boe-Bot, el podem aconseguir de manera gratuïta a la pàgina web de Parallax [6].

## 3.2. Programa de control

Per a poder gestionar el moviment en l'entorn de treball del robot, és necessari el programa de control que tingui en compte les possibles situacions esmentades en el punt 3.1. Abans de gestionar les situacions conflictives però, cal tenir ben definit el programa bàsic de seguidor de línies que serà l'encarregat de moure el robot per l'entorn de treball. Per tant el programa de control seguirà la lògica de comprovar si ens trobem en una de les situacions especials que hauran de gestionar-se amb una rutina diferent, i sino seguirà el control de seguir la línia. És a dir, el robot seguirà la línia fins que aparegui un esdeveniment que provoqui un canvi en el seu comportament.

Com s'ha comentat anteriorment, el robot seguirà un cicle infinit que es basarà en controlar els sensor, decidir quina acció correspon i finalment actuar sobre els motors. Aquest cicle a mode de pseudocodi a nivell molt abstracte és pot implementar de la següent manera:

```
//----- Rutina principal -----//

DO

    GOSUB Control_sensors

    GOSUB Enviar_motors

LOOP

//----- Subrutina – Control_sensors -----//

Control_sensors:

servoEsquerra = Obtenir_dades_sensors

servoDreta = Obtenir_dades_sensors


RETURN

//----- Subrutina – Enviar_motors -----//

Enviar_motors:

PULSEOUT 13, servoEsquerra

PULSEOUT 12, servoDreta

RETURN
```

Amb el pseudocodi anterior queda definit l'esquelet del programa que s'ha d'implementar per controlar els moviments del robot en l'entorn de treball. S'ha fer servir un llenguatge molt semblant al PBASIC que és el llenguatge que s'utilitzarà per a la programació del microcontrolador.

La subrutina de "Control\_sensors" es desglossarà amb les diferents subrutines encarregades de detectar i gestionar els esdeveniments "especials" definits en el punt 3.1.

A continuació es defineixen les rutines que seran les encarregades de portar el control del mini-robot (AGV) dins de l'entorn de treball.

### 3.2.1. Subrutina: Enviar motors

Aquesta subrutina és una de les bàsiques en el programa de control del moviment del mini robot, ja que és l'encarregada de enviar la informació necessària als servos de cada roda per realitzar el moviment, és a dir, és la rutina encarregada de moure el robot en si, amb els paràmetres necessaris.

Aquest paràmetres seran variables dins del programa que és modificaran en funció de la situació que es trobi el mini robot.

La connexió dels servos és realitza per els pins 12 i 13 (figura 3.1) de la placa del robot essent així el pin 12 l'encarregat de moure el servo de l'esquerra (roda esquerra) i el pin 13 l'encarregat de moure el servo de la dreta (roda dreta).

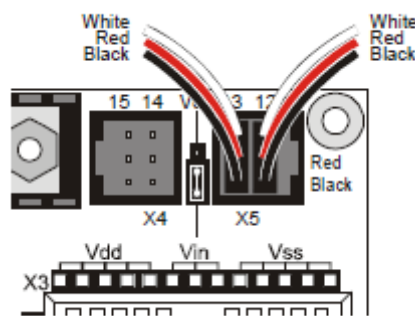


Figura 3.1. Connexió dels servos a la placa del robot.[5]

Per a poder moure els servos, la instrucció de BASIC Stamp que ens permet fer això es el PULSEOUT, que envia un pols al pin indicat de la placa amb un segon argument (Duration) que indica la durada d'aquest pols:

**PULSEOUT** "pin", "Duration"

Inicialment cal ajustar els servos a un pols de 1.5 milisegons que serà el pols necessari per a tenir aturats els servos. En el manual del Boe-bot [5] trobem la manera de definir l'argument "Duration" ja que cada pols ha de ser dividit per  $2 \mu s$ , per tant l'argument que manté els servos aturats surt del següent càlcul:

$$Duration = \frac{1.5ms}{2 \mu s} = \frac{0.0015s}{0.000002s} = 750$$

Per tant, per tenir el robot aturat calen les següents instruccions:

**PULSOUT 13, 750**

**PULSOUT 12, 750**

Llavors per a fer rotar els servos cap a la dreta (sentit horari) cal reduir el paràmetre "Duration" de 750 i per fer rotar els servos a l'esquerra (sentit anti-horari) cal augmentar el valor de "Duration".

Per tant, per a fer avançar el robot, cal moure els servos a la mateixa velocitat, degut a que els servos estan muntats de forma simètrica, les funcions PULSOUT, quedarien de la següent manera:

**PULSOUT 13, 650**

**PULSOUT 12, 850**

No obstant això aquestes dues instruccions s'han de repetir un seguit de vegades per aconseguir un moviment constant, per això el programa principal és basa en un bucle infinit.

Per a que aquesta subrutina pugui variar en funció del entorn, cal definir dues variables globals que aniran sent modificades per altres rutines per a obtenir el valor de "Duration" necessari en cada moment. Per tant la subrutina, quedaria de la següent manera:

servodreta **VAR Word**

servoesquerra **VAR Word**

Moure:

**PULSOUT 13, servodreta**

**PULSOUT 12, servoesquerra**

**RETURN**

### 3.2.2. Subrutina: Parada per obstacle

La rutina de parada per obstacle, és la responsable d'evitar possibles col·lisions del AGV en l'entorn que es mogui, aturant el mini robot quan es detecti un obstacle en la seva trajectòria.

Per a implementar aquesta subrutina és necessari un sensor de Ultrasons per al Boe-bot, aquest sensor treballa a una freqüència de 40 kHz, essent ultrasons aquelles freqüències superiors als 20 kHz. Aquest sensor és el que es mostra a la següent figura:



Figura 3.2. "The Ping ))) Sensor" (Sensor d'ultrasons per a Boe-bot).[7]

Aquest sensor funciona a mode de sonar, enviant un ultrasò i captant el temps que triga en retornar la seva ona emesa, per tant, capta el pols en qüestió i l'envia al microcontrolador de la placa del Boe-Bot com il·lustra la següent figura:

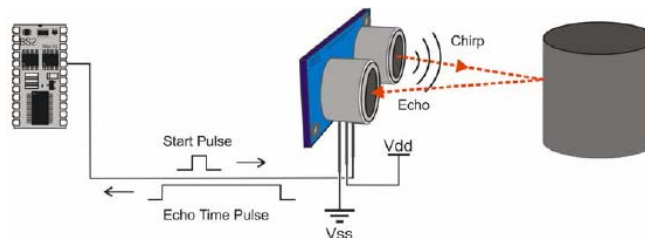


Figura 3.3. Funcionament del sensor d'ultrasons[7]

Per tant, amb la funció PULSEOUT (i amb el càlcul del paràmetre "Duration" definit en el punt 3.2.1) podem definir un pols de 10  $\mu$ s per a la "ona emissora" i amb la instrucció PULSEIN recollir la "ona rebotada" i guardar-a en una variable que anomenarem "temps". En aquest cas, connectem el sensor d'ultrasons al pin 15 de la placa del Boe-bot.

Degut a que tenir les dades en temps pot dificultar els càlculs, es convenient transformar aquest temps en que triga la ona a ser rebotada en la distància real al obstacle, per a poder-ho fer fem servir l'operador \*\* de Basic Stamp juntament amb una constant (2251) per a poder convertir el temps a centímetres [7], i així tenir una variable "distancia" amb la mesura en centímetres que hi ha fins al següent obstacle.

D'aquesta manera, si el robot s'apropa a una distància de possible col·lisió modificarem les variables globals del control dels servos per a que s'aturi el mini robot.

La subrutina de "Parada per obstacles" quedarà per tant, de la següent manera:

time **VAR Word**

distancia **VAR Word**

Obstacle:

**PULSOUT** 15, 5

**PULSIN** 15, 1, time

distancia = time \*\* 2251

**IF** (distancia) < 20 **THEN**

servodreta = 750

servoesquerra = 750

**ENDIF**

**RETURN**



### 3.2.3. Subrutina: Bifurcació

La funcionalitat del mini-robot per a poder gestionar una bifurcació en el seu recorregut es quelcom necessària en tant que a partir de les bifurcacions es simulen les balises de l'escenari emulat. Per tant cada vegada que el mini-robot gestioni una bifurcació voldrà dir que en el programa de control en aquell moment haurà d'anar a una de les estacions situades a les balises, com es mostra en el punt "2.3. Àrea de treball".

Per tant, abans de indicar-li al servos del mini-robot el gir que haurà de realitzar, cal primerament diferenciar d'alguna manera que una bifurcació serà agafada. Per a poder dur el control de les bifurcacions es disposa d'unes marques laterals a la línia que fa servir el mini-robot per navegar per l'escenari per a indicar que s'aproxima una bifurcació.

Tanmateix, per a poder obtenir dades d'aquestes marques cal una tercera parella de emissor/receptor de infraroig per a detectar aquestes marques situades al lateral de la línia principal i que indiquen una possible bifurcació.



Figura 3.4. Marca de bifurcació

Una vegada llegida la marca de possible i bifurcació i havent comprovat si s'ha d'agafar o no la bifurcació, en cas d'agafar-la es procedirà de la següent manera:

Al detectar la marca de bifurcació caldrà anul·lar durant un període d'uns 3 segons el sensor infraroig de seguiment de línia de la banda dreta, d'aquesta manera, només tenint actiu el infraroig de l'esquerra s'aconsegueix que només "existeixi" el camí de la bifurcació per al que fa els sensor d'infraroig (encarregats de mantenir el mini-robot sobre la línia). Així una vegada el mini-robot comença a virar cap a l'esquerra, ja que es l'únic camí possible en aquell moment, i ja estigui sobre el camí bifurcat, és tornar a activar el sensor de línia de la dreta.

Per tant, la subrutina que gestiona la bifurcació cap a un dels objectius a assolir, cal que estigui en funcionament durant un període d'uns 3 segons, temps suficient per a realitzar el viratge cap a la estació objectiu.

Així doncs és tracta d'un codi similar a la rutina que manté el robot sobre la línia, però en aquest cas el càlcul proporcional només es realitza sobre el servo de l'esquerra que serà l'encarregat de gestionar el gir, mentre que el servo de la dreta actuarà de manera fixa durant 3 segons.

Degut al funcionament del programa la rutina que fa moure els servos disposa d'una pausa de 5 milisegons per evitar que el programa corri més que el moviment físic del robot, aprofitant aquesta pausa, es pot ajustar el delay de 3 segons mitjançant un bucle.

Una vegada finalitzat els 3 segons de gir, la subrutina tornarà al control de navegació bàsic de seguir línia i passarà a gestionar la possible estació objectiu.

anular\_dreta **VAR Bit**  
 temps\_girar **VAR Word**

```

IF (anular_dreta = 1) THEN
  FOR temps_girar = 0 TO 15000

    distanceLeft = 0
    FOR freqSelect = 0 TO 4
      LOOKUP freqSelect,[37500,38250,39500,40500,41500], irFrequency
      FREQOUT 10,1,irFrequency

    irDetectLeft = IN9
    distanceLeft = distanceLeft + irDetectLeft
  NEXT

  pulseLeft = SetPoint - distanceLeft * Kpl + CenterPulse
  pulseRight = 750
  PULSOUT 12,pulseLeft
  PULSOUT 13,pulseRight
  PAUSE 5

  NEXT
ENDIF

```

En el cas que una vegada detectada la marca de possible bifurcació no s'hagi d'agafar aquesta, és a dir, que la estació actual on es trobi el robot no forma part dels objectius, caldrà gestionar la bifurcació de diferent manera.

De la mateixa manera que quan s'havia de girar, l'estratègia en aquest cas és similar, ja que en el cas de que no s'hagi d'anar cap a l'estació, el que cal fer es continuar recte durant el període de temps de 3 segons. Així doncs com en el cas anterior s'anul·laven els sensors de marca de la banda dreta, obligant al robot a només reconèixer la corba, ara s'anul·laran els sensors de marca de banda esquerra, obligant així al robot a creure que la corba no existeix i que per tant ha de seguir recte.

Per tant el codi, haurà de mantenir el rumb que portava el robot abans d'arribar a la bifurcació i mantenir-lo durant 3 segons.

anular\_dreta **VAR Bit**  
temps\_girar **VAR Word**

```
IF (anular_esquerra = 1) THEN  
  FOR temps_girar = 0 TO 15000  
  
    pulseLeft = SetPoint - distanceLeft * Kpl + CenterPulse  
    pulseRight = SetPoint - distanceRight * Kpr + CenterPulse  
    PULSOUT 12,pulseLeft  
    PULSOUT 13,pulseRight  
    PAUSE 5  
  
  NEXT  
ENDIF
```

### 3.2.4. Subrutina: Unió

Degut a la configuració de l'àrea de treball, aquesta subrutina haurà de ser cridada i gestionada de diferents maneres ja que hi ha dos casos d'unió en l'àrea de treball.

En el primer cas d'unió, aquesta subrutina només serà cridada en el cas de que s'hagi agafat prèviament una bifurcació (la qual condueix a una de les balises programables com a estació). Com succeeix en el cas de la bifurcació per a poder saber que el mini-robot s'aproxima a una unió, caldrà detectar una marca lateral prèvia a que els servos comencin a realitzar el gir.

En el cas de la unió (post-bifurcació) la marca lateral es trobarà com mostra la següent figura:



Figura 3.5. Marca d'unió post-bifurcació

Per a realitzar el gir i recuperar la línia principal és seguirà el següent procediment:

Al detectar la marca d'unió caldrà anul·lar durant un període d'uns 3 segons el sensor infraroig de seguiment de línia de la banda dreta, d'aquesta manera, només tenint actiu el infraroig de l'esquerra s'aconsegueix que només "existeixi" el camí de la unió per al que fa els sensor d'infraroig (encarregats de mantenir el mini-robot sobre la línia). Així una vegada el mini-robot comença a virar cap a l'esquerra, ja que es l'únic camí possible en aquell moment, i ja estigui sobre el camí principal altre cop, és tornar a activar el sensor de línia de la dreta.

Per tant una vegada el mini-robot està sobre el camí de la bifurcació torna a tenir els sensors encarregats de la navegació actius.

Com en el cas de la bifurcació el codi tractarà d'anul·lar els sensors de la dreta del robot per fer creure que només hi ha un camí possible durant 3 segons.

anular\_dreta **VAR Bit**  
 temps\_girar **VAR Word**

```

IF (anular_dreta = 1) THEN
  FOR temps_girar = 0 TO 15000

    distanceLeft = 0
    FOR freqSelect = 0 TO 4
      LOOKUP freqSelect,[37500,38250,39500,40500,41500], irFrequency
      FREQOUT 10,1,irFrequency

      irDetectLeft = IN9
      distanceLeft = distanceLeft + irDetectLeft
    NEXT

    pulseLeft = SetPoint - distanceLeft * Kpl + CenterPulse
    pulseRight = 750
    PULSOUT 12,pulseLeft
    PULSOUT 13,pulseRight
    PAUSE 5

  NEXT
ENDIF

```

El segon cas d'unió gestiona la situació en que el mini-robot es troba per la línia principal i per la seva esquerra arriba el camí d'una balisa que no tocava entrar-hi. Com succeeix en l'altre cas d'unió caldrà que la tercera parella de emissor/receptor d'infraroig detecti una marca lateral per a gestionar la unió.



Figura 3.6. Marca d'unió des de camí principal

Per a poder seguir pel camí principal sense perill de que el mini-robot agafes la unió en direcció contraria caldrà el següent procediment:

Al detectar la marca d'unió caldrà indicar als servos del mini-robot les dades necessàries per a continuar recte pel camí principal el temps que duri la unió. Per tant, en aquest cas, el sistema de sensors infraroig que manté el robot sobre la línia i que ajusta els servos en conseqüència serà passat per alt, donant-li d'altre banda als servos les revolucions necessàries per a continuar recte.

Així doncs, una vegada s'ha acabat el temps de la unió, el mini-robot tornarà a activar les rutines de seguiment de línia.

Com en el cas de la bifurcació que calia mantenir el rumb que es portava anteriorment, en aquest cas també es mantindrà el rumb anterior durant un període de 3 segons.

anular\_dreta **VAR Bit**  
temps\_girar **VAR Word**

```

IF (anular_esquerra = 1) THEN
  FOR temps_girar = 0 TO 15000

    pulseLeft = SetPoint - distanceLeft * Kpl + CenterPulse
    pulseRight = SetPoint - distanceRight * Kpr + CenterPulse
    PULSOUT 12,pulseLeft
    PULSOUT 13,pulseRight
    PAUSE 5

  NEXT
ENDIF

```

### 3.2.5. Subrutina: Parada de programació

En aquesta rutina el mini-robot esperarà els objectius, enviats a través de l'ordinador via bluetooth, de quines balises són les quals ha d'anar, per tant, quan s'arribi en aquest punt, el robot romandrà quiet rebent informació de quins objectius ha d'assolir, i esperarà com a última ordre la que marqui que pot començar a realitzar els objectius enviats.

Per a poder establir comunicació amb el mini-robot, ens cal el mòdul de bluetooth "Embedded blue eb500"[8] (com el que es mostra en la següent imatge) ja que disposa de connexió directa amb la placa del mini-robot. Aquest mòdul fa servir els pins P0, P1, P5 i P6 que hauran de quedar lliures d'altres usos.



Figura 3.7. Mòdul Bluetooth eb500 sobre la placa del mini-robot [8]

Primerament caldrà establir connexió amb l'ordinador. Afortunadament, la majoria d'ordinadors portàtils disposen d'un mòdul de bluetooth. Per tant, només caldrà muntar el mòdul a la placa, donar-li energia, i activar el mòdul de bluetooth del ordinador. A través del sistema operatiu del ordinador, és detecta el mòdul bluetooth i s'estableix connexió amb la clau "0000". En aquest punt, des del software de gestió de bluetooth del sistema operatiu del ordinador, s'obté el port des del qual s'ha establir la comunicació, en aquest cas el "COM7".

Lavors des d'un programa de Telnet, en aquest cas Hyperterminal de Microsoft, és realitza una connexió a través del port "COM7", i el mini-robot ja estarà preparat per a rebre els objectius a assolir.

Per tant, per a poder interpretar els objectius que es donen des de l'ordinador, caldrà el següent codi:

```

marca_objectiu CON %1
index VAR Nib
objectius VAR Byte
vector_objectius VAR Nib(8)

```

```
index = 0
```

**Llegir:**

```
SERIN 0,84,[STR objectius\1]
```

```
FOR index = 0 TO 7
```

```
IF (objectius = index) THEN
```

```
vector_objectius (index) = marca_objectiu
```

```
IF (objectius = 9) THEN
```

```
vector_objectius (7) = 9
```

```
GOTO Fi_objectius
```

```
ENDIF
```

```
ENDIF
```

```
NEXT
```

```
index = 0
```

```
GOTO Llegir
```

```
Fi_objectius:
```

```
RETURN
```

Com s'explica en el punt "2.3 Àrea de treball" el robot rebrà una seqüència d'objectius finalitzades amb el nombre 9.

El codi anterior, espera a que l'usuari introdueixi els objectius a on ha d'anar el robot i els anirà posant en un vector que simbolitzarà les balises a on ha d'anar a el robot, per tant, si s'introdueix la estació numero 3, s'afegirà un 1 a la posició numero tres del vector esperant sempre el numero 9 que indica el fi d'instruccions, en aquest cas, la codificació del numero 9 es posarà a l'última posició del vector, concretament a la posició 8.

Per exemple, si s'introdueix la seqüència d'instruccions 3-5-2-9, el vector resultant serà el següent:

0	1	1	0	1	0	0	9
---	---	---	---	---	---	---	---

Així doncs, quan el robot es trobi davant d'una bifurcació només haurà de comprovar si el nombre d'aquella bifurcació, corresponent a una de les balises o estacions de treball, té existència, correspon a igual a 1, en la posició del vector d'objectius, i així sabrà si ha d'agafar la bifurcació cap a la balisa o no.



### 3.2.6. Subrutina: Recirculació

La subrutina de recirculació és un cas especial de bifurcació. En aquest cas es troba en el final del circuit de l'entorn de treball. Al trobar la marca lateral de "recirculació", s'ha de comprovar quins objectius han estat assolits fins al moment. En el cas de que encara restin objectius per complir, el mini-robot agafarà aquesta bifurcació i així podrà tornar a realitzar el circuit per assolir els objectius que encara no s'hagin complert.

Per altre banda, si ja estan tots els objectius assolits, el mini-robot seguirà endavant per a poder acabar arribant a una nova "parada de programació".

Així doncs, aquesta rutina ha d'implementar una comprovació de si encara queden objectius per assolir i després cridar a la subrutina de bifurcació, segons si s'ha d'agafar la bifurcació o no.

Per realitzar si la comprovació d'objectius s'ha assolit caldrà fer ús del vector d'objectius creat per a rebre instruccions, i només caldrà fer una suma dels  $n - 1$  elements del vector comprovant si aquesta resulta 0, ja que l'últim element del vector, està marcat amb un 9 com a final de rebre instruccions.

En el cas de que tots els objectius hagin sigut assolits caldrà posar a 0 la última posició del vector (que contenia el 9 com a marca de fi d'instrucció) i dirigir el robot cap a la primera parada de programació, tornant així a començar el cicle d'estats com si fos la primera vegada.

Per contra, si encara li manquen objectius, es reconduirà altre cop cap a la primera balisa indicant-li així que la següent balisa que és trobarà serà la primera.

```

acaba VAR Byte
vector_objectius VAR Nib(8)
control_marques VAR Byte
anular_dreta VAR Bit
anular_esquerra VAR Bit
index VAR Nib

acaba = 0

FOR index = 0 TO 6
  acaba = acaba + vector_objectius (index)
NEXT

IF (acaba = 0) THEN

  vector_objectius (7) = 0
  control_marques = 0
  anular_esquerra = 1
  GOSUB Send_pulse

```

**ELSE**

control\_marques = 1

anular\_dreta = 1

**GOSUB** Send\_pulse

**ENDIF**

**RETURN**

Una vegada s'ha decidit si es pren o no la bifurcació, cal cridar la subrutina que gestiona la bifurcació amb la estratègia comentada en el punt "3.2.3. Subrutina: Bifurcació" mantenint el rumb del robot si els objectius s'han assolit o agafant la bifurcació en cas contrari anul·lant els sensors de la dreta durant el període de 3 segons.

### 3.2.7. Subrutina: Seguir línia

Aquesta rutina és una de les més importants en el control del moviment del mini-robot en l'entorn, ja que és l'encarregada de controlar, que el mini-robot circuli per la via marcada que delimita el circuit que realitza per dur a terme la seva tasca.

Per tant, aquesta rutina serà capaç de mantenir el rumb del mini-robot sobre el rail de l'entorn de treball. Per a dur a terme aquesta tasca, es necessiten els sensors d'infraroig de detecció d'obstacles, ja que en aquest cas, la cinta de vinil que marca l'àrea de treball és detectada com a "obstacle".

Per a que pugui seguir la línia de vinil essent aquesta considerada "obstacle", és necessiten dos sensors d'infraroig, un que controlarà la part dreta i l'altre la part esquerra. Així doncs, quan el robot es pugui desviar de la línia, un dels sensors començarà a notar la línia, i al considerar-la un obstacle, adreçarà el rumb del mini-robot cap a la direcció contrària, quedant així altre vegada sobre la línia.

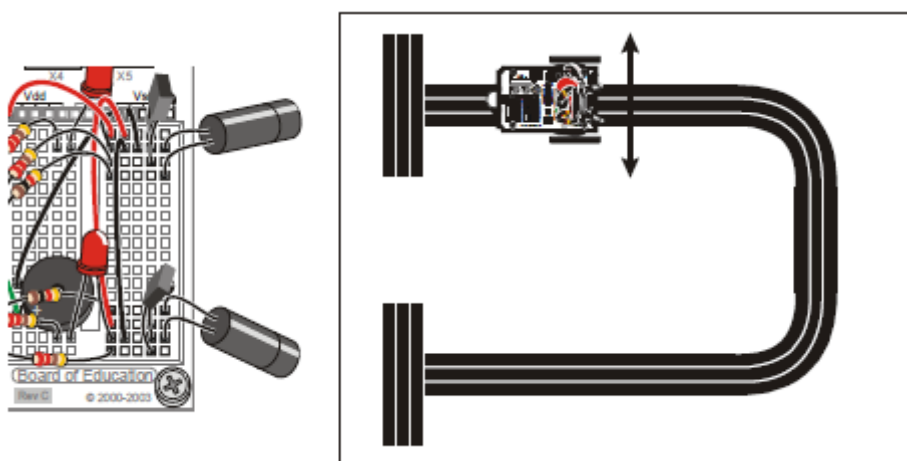


Figura 3.7. Orientació i comportament dels sensors d'ultrasons[5]

Per tant, els sensors només veuran la línia quan el robot es desviï de la seva direcció, i el seu rumb quedarà corregit.

En aquest cas, per a que els sensors d'infraroig detectin obstacles, cal fer-los funcionar amb diferents freqüències en funció de la distància a on es vol trobar l'obstacle (figura 3.5). Per tant cal anar fent un mostreig de les diferents freqüències d'una manera iterativa i constant. Per a poder realitzar això, la funció LOOKUP del BasicStamp ens permet fer el mostreig en una estructura FOR i guardar el valor de la freqüència determinada en una variable per a poder-la manipular. Així doncs, una vegada detectada la línia per un dels dos sensors (cas en el que se

surt de la línia o ve una corba) caldrà rectificar la velocitat del motor oposat al sensor, per a tornar a posicionar el mini-robot sobre la línia, o bé que traci la corba correctament.

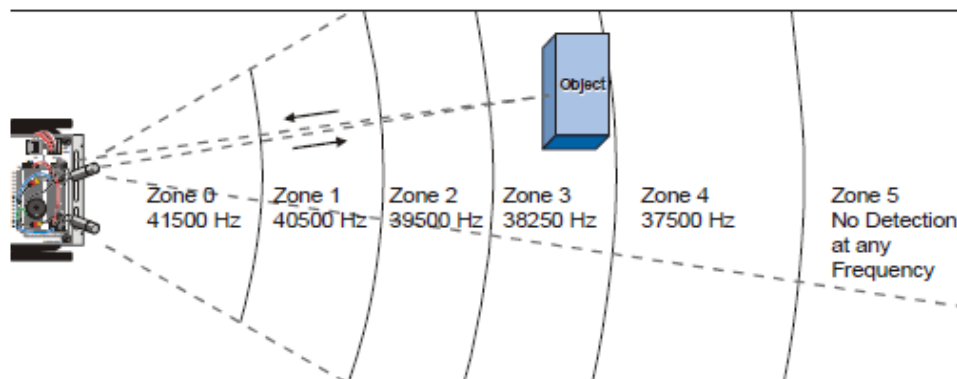


Figura 3.8. Freqüències i zones des sensors IR[5]

Així doncs, aquesta subrutina modificarà les variables encarregades de guardar el valor que s'envia als motors per a moure el robot, sempre que el robot marxi del rumb que té estipulat.

Per implementar aquesta rutina caldran algunes constants que representaran valors necessaris per calcular el factor de correcció del moviment del mini-robot.

Per tant, la subrutina de seguir línies queda de la següent manera:

' -----[ Constants i variables ]-----'

Kpl **CON** 35

Kpr **CON** -35

SetPoint **CON** 3

Centrar **CON** 750

freqSelect **VAR** Nib

irFrequencia **VAR** Word

irDetectEsquerra **VAR** Bit

irDetectDreta **VAR** Bit

distanciaEsquerra **VAR** Nib

distanciaDreta **VAR** Nib

Seguir\_linia:

distanciaEsquerra = 0

distanciaDreta = 0

**FOR** freqSelect = 0 **TO** 4

**LOOKUP** freqSelect,[37500,38250,39500,40500,41500], irFrequencia

**FREQOUT** 8,1,irFrequencia

irDetectEsquerra = IN9

distanciaEsquerra = distanciaEsquerra + irDetectEsquerra

servoEsquerra = SetPoint - distanciaEsquerra \* Kpl + Centrar

**FREQOUT** 2,1,irFrequencia

irDetectDreta = IN0

distanciaDreta = distanciaDreta + irDetectDreta

servoDreta = SetPoint - distanciaDreta \* Kpr + Centrar

**NEXT**

**RETURN**

### 3.2.8. Rutina Principal

Una vegada definides totes les subrutines encarregades de gestionar les diferents situacions en que es pot trobar el robot, cal muntar el programa principal encabint aquests subrutines a nivell de mòduls i interconnectar-los entre si.

Com es mostrava al principi d'aquesta secció, es manté el flux bàsic conceptual del moviment del robot, que a grans trets es pot definir com un cicle infinit a on trobem:

Obtenció de dades dels sensors → Presa de decisions → Actuació/moviment del robot

Per a que el robot sàpiga en tot moment a on es troba dintre del mapa d'actuació, és necessari mantenir un control d'estats per a saber a on es troba en tot moment. Per tant el codi que és mostra a continuació està preparat exclusivament per l'àrea de treball definida, ja que a partir de les marques laterals que es va trobant el robot, s'ubica en el mapa per a reconèixer les estacions/balises que ha de tractar.

Sent necessari la connexió entre les subrutines descrites anteriorment, en el programa principal hi ha variables de control i subrutines de control que assegurin el flux correcte del programa creant així en alguns casos subrutines en cascada o niuades.

Així doncs el codi que es mostra a continuació és l'encarregat de controlar el robot en tot moment i tenir coneixement del mapa o àrea de treball a on es mou, rebre els objectius i traçar el camí per dur-los a terme.

El codi següent està en el format de Basic Stamp.

```

' {$STAMP BS2} ' Directiva Stamp.
' {$PBASIC 2.5} ' Directiva PBASIC.
' -----[ Títol ]-----
'
' PFC - AGV - Simulació planta mèdica - programa de control final
'
' -----[ Constants ]-----

Kpl CON 35 ' Canvi de -35 to 35
Kpr CON -35 ' Canvi de 35 to -35
SetPoint CON 1 ' Canvi de 2 to 3.
marca_objectiu CON %1 'per marcar a l'array d'objectius
CenterPulse CON 750

' -----[ Variables ]-----

freqSelect VAR Nib
irFrequency VAR Word
irDetectLeft VAR Bit
anular_dreta VAR Bit
anular_esquerra VAR Bit
irDetectCentre VAR Bit
irDetectRight VAR Bit
distanceLeft VAR Nib
distanceCentre VAR Nib
distanceRight VAR Nib
pulseLeft VAR Word
pulseRight VAR Word
temps_girar VAR Word
objectius VAR Byte
time VAR Word
index VAR Nib
vector_objectius VAR Nib(8)
control_marques VAR Byte

' -----[ Inicializació ]-----

FREQOUT 4, 2000, 3000
control_marques = 0
anular_esquerra = 0
anular_dreta = 0

' -----[ Rutina Principal ]-----

DO
GOSUB Get_Ir_Distances
' Calcula sortida proporcional del servos segons led
pulseLeft = SetPoint - distanceLeft * Kpl + CenterPulse
pulseRight = SetPoint - distanceRight * Kpr + CenterPulse
GOSUB Obstacle
GOSUB Send_Pulse
LOOP

```

```

' -----[ Subrutina - Get IR Distances ]-----

Get_Ir_Distances:
distanceLeft = 0
distanceRight = 0
distanceCentre = 0
FOR freqSelect = 0 TO 4
LOOKUP freqSelect,[37500,38250,39500,40500,41500], irFrequency
FREQOUT 10,1,irFrequency
irDetectLeft = IN9
distanceLeft = distanceLeft + irDetectLeft
FREQOUT 8,1,irFrequency
irDetectCentre = IN7
distanceCentre = distanceCentre + irDetectCentre
FREQOUT 2,1,irFrequency
irDetectRight = IN0
distanceRight = distanceRight + irDetectRight

IF (distanceRight>2) THEN
' salt a subrutina que controla si s'ha de girar o no

GOSUB Controla_gir
control_marques = control_marques + 1

ENDIF

NEXT
RETURN

' -----[ Subrutina - Obstacle ]-----

Obstacle:
PULSOUT 15, 5
PULSIN 15, 1, time
time = time ** 2251
DEBUG CR, "Distance = ", DEC4 time, " cm" 'transformació de distancia a cm
IF (time) < 20 THEN ' si menor de 10 cm atura
pulseRight = 755
pulseLeft = 739
ENDIF
RETURN

' -----[ Subrutina - Send Pulse ]-----

Send_Pulse:

IF (anular_dreta = 1) THEN
FOR temps_girar = 0 TO 15000 ' 3 segons

distanceLeft = 0
FOR freqSelect = 0 TO 4
LOOKUP freqSelect,[37500,38250,39500,40500,41500], irFrequency
FREQOUT 10,1,irFrequency

irDetectLeft = IN9
distanceLeft = distanceLeft + irDetectLeft
NEXT

```

```

pulseLeft = SetPoint - distanceLeft * Kpl + CenterPulse
pulseRight = 750
PULSOUT 12,pulseLeft
PULSOUT 13,pulseRight
PAUSE 5

NEXT
ENDIF

IF (anular_esquerra = 1) THEN
  FOR temps_girar = 0 TO 15000      ' 3 segons

    pulseLeft = SetPoint - distanceLeft * Kpl + CenterPulse
    pulseRight = SetPoint - distanceRight * Kpr + CenterPulse
    PULSOUT 12,pulseLeft
    PULSOUT 13,pulseRight
    PAUSE 5

  NEXT
ENDIF

anular_dreta = 0
anular_esquerra = 0

PULSOUT 12,pulseLeft
PULSOUT 13,pulseRight
PAUSE 5
RETURN

' -----[ Subrutina - Controla gir ]-----

Controla_gir:
SELECT control_marques
CASE 0
  GOSUB Rebre_objectius
CASE 5
  vector_objectius (7) = 0
  GOSUB Rebre_objectius
CASE 6
  GOSUB Recirculacio
CASE ELSE
  IF (vector_objectius (control_marques) = 1) THEN

    anular_dreta = 1
    GOSUB Gestiona_estacio
    anular_dreta = 1
    GOSUB Send_pulse

  ELSE
    anular_esquerra = 1
    GOSUB Send_pulse
  ENDIF
END
ENDSELECT
RETURN

```



```

' -----[ Subrutina - Gestiona estacio ] -----

Gestiona_estacio:

vector_objectius (control_marques) = 0

DO
distanceLeft = 0
distanceRight = 0
distanceCentre = 0
FOR freqSelect = 0 TO 4
LOOKUP freqSelect, [37500,38250,39500,40500,41500], irFrequency
FREQOUT 10,1,irFrequency
irDetectLeft = IN9
distanceLeft = distanceLeft + irDetectLeft
FREQOUT 8,1,irFrequency
irDetectCentre = IN7
distanceCentre = distanceCentre + irDetectCentre
FREQOUT 2,1,irFrequency
irDetectRight = IN0
distanceRight = distanceRight + irDetectRight

NEXT

pulseLeft = SetPoint - distanceLeft * Kpl + CenterPulse
pulseRight = SetPoint - distanceRight * Kpr + CenterPulse
GOSUB Obstacle
GOSUB Send_Pulse

IF (distanceRight>2) THEN
RETURN
ENDIF

LOOP

' -----[ Subrutina - rebre objectius ] -----

Rebre_objectius:

index = 0

Llegir:
SERIN 0,84,[STR objectius\1]

FOR index = 0 TO 7
IF (objectius = index) THEN
vector_objectius (index) = marca_objectiu

IF (objectius = 9) THEN
vector_objectius (7) = 9
GOTO Fi_objectius
ENDIF

ENDIF
NEXT

```

```

index = 0
GOTO Llegir

Fi_objectius:
RETURN

' ----- [ Subrutina - Recirculacio -----

Recirculacio:

acaba VAR Byte
acaba = 0

FOR index = 0 TO 6
acaba = acaba + vector_objectius (index)
NEXT

IF (acaba = 0) THEN 'objectius finalitzats

vector_objectius (7) = 0
control_marques = 0
anular_esquerra = 1
GOSUB Send_pulse

ELSE

control_marques = 1
anular_dreta = 1
GOSUB Send_pulse

ENDIF

RETURN

```

# Capítol 4. Proves i errors

En aquest capítol és descriuen les proves o casos de test que s'han dut a terme pel funcionament final del robot, així com les traves o errors que anaven sortint a mesura que es desenvolupava el software de control del robot.

## 4.1 Proves i casos de test

Degut a la planificació del projecte, s'han anat realitzant proves durant el desenvolupament del mateix, ja que en molts dels casos no es podia seguir avançant fins que no es donava per testejat i aprovat la part de software del programa de control anterior a la fita.

D'aquesta manera és pot dir que s'han fet proves per cada subrutina en particular de les que conforma el programa de control final i anaven fent créixer i adaptant el programa de control.

Es poden separar a grans trets en les que gestionen el moviment i les que controlen el rumb i les instruccions, essent necessari implementar primer les de gestió de moviment abans de les de control de rumb.

En el cas de la subrutina de control d'obstacles, es va desenvolupar de les primeres sense encara tenir el seguidor de línies, així doncs el robot es movia sense rumb però es buscava que reaccions davant d'un obstacle aturant-se, ja que si això succeïa amb un rumb aleatori, aquesta subrutina funcionaria bé quan s'implementés dins del programa de control en un entorn de seguidor de línia.

Una altre de les rutines de moviment, i potser una de les més bàsiques en el control final del robot, es la de seguir línia, ja que és el primer pas des del rumb aleatori a poder gestionar un rumb guiat. Aquesta subrutina es va implementar i testejar per separat amb només dos parelles d'emissor/receptors d'infraroig (sense detector de marques), i amb una àrea de treball sense bifurcacions ni unions, sent simplement un circuit tancat i circular. Veient així que es mantenia bé sobre la línia i ja era reactiu a obstacles amb la integració de codi de la subrutina testejada anteriorment (detecció d'obstacles).

Una vegada operatives les rutines de moviment bàsic (seguir una línia i aturar-se davant d'obstacle) és va introduir la tercera parella emissors/receptors d'infraroig, per a poder detectar marques i gestionar bifurcacions i unions dins del circuit de test. Aquestes bifurcacions i unions encara no tenien sentit aparent, només es provaven per gestionar el moviment.

És en aquest moment quan és construeix l'entorn de treball final, amb el circuit amb totes les seves marques i comença la part de proves de les rutines de gestió de rumb i instruccions.

Primerament, és van realitzar proves amb un rumb marcat en diferents estacions del mapa de treball però sense la comunicació per afegir objectius. Per així comprovar que el robot era capaç de moure's per l'entorn de treball i entrar dins de les balises o estacions de treball.

Una vegada la navegació pel mapa de treball no mostrava errors, es va desenvolupar finalment la rutina de comunicació per a enviar els objectius al robot i a la que aquesta estava operativa, es va ampliar el programa principal amb la lògica necessària per tenir codificat l'entorn de treball, fent així coneixedor al robot del espai a on s'ha de moure.

En aquest punt, només calia provar casuístiques de funcionament per a veure si el programa de control final era estable en els casos que es poguessin donar.

A continuació es presenta el flux d'execució d'un cas de prova concret, que ha servit per provar que totes les subrutines mencionades anteriorment són utilitzades i funcionen correctament. És per tant un cas a on es mostren els possibles casos en que es trobaria el robot.

Per aconseguir un funcionament de totes les rutines, caldrà primerament encendre el robot i el mòdul de bluetooth del sistema operatiu per aconseguir comunicació entre l'ordinador i el robot mitjançant l'hyperterminal de Microsoft.

Una vegada s'ha establert la connexió, posarem el robot sobre la línia en la posició prèvia a la primera parada de programació, que és el lloc on es rebran els objectius. En el moment en que el robot arranca el seu moviment, s'aturarà a la primera parada de programació i allí li enviarem els següents objectius: 1-3-9

D'aquesta manera el vector d'objectius quedarà de la següent manera:

1	0	1	0	0	0	0	9
---	---	---	---	---	---	---	---

Indicant així que el robot ha de complir els objectius situats a les balises 1 i 3.

El robot anirà seguint la línia fins arribar a la marca lateral corresponent a la primera balisa, quan el sensor detecti aquesta marca, comprovarà que el vector d'objectius tingui un "1" a la primera posició, fet que farà de decisió en una de les dues opcions següents:

1	0	1	0	0	0	0	9
---	---	---	---	---	---	---	---

- **Entrar a la rutina de "gestionar estació" i girar a la esquerra cap a la balisa.**
- Seguir recte deixant de llarg la balisa.

En aquest cas, la opció marcada amb negreta és la que s'agafarà degut a que la balisa 1 esta com a objectiu. Per tant, s'actualitzarà la variable encarregada de conèixer el numero de balisa a la que es troba el robot, s'ignoraran els sensors de moviment de la part dreta per poder gestionar el gir, una vegada fet el gir, és recuperarà la rutina de moviment bàsic de seguiment de línia i s'actualitzarà el vector objectius de la següent manera:

<b>0</b>	0	1	0	0	0	0	9
----------	---	---	---	---	---	---	---

D'aquesta manera queda marcat que el robot ja ha realitzat l'objectiu 1 i segueix el seu rumb, fins a trobar la següent marca lateral, que en aquest cas, està marcant que s'acaba la balisa i s'aproxima una unió.

Arribat aquest punt, el robot torna a gestionar el gir ignorant els sensors drets, per a recuperar el camí principal de l'entorn de treball.

Seguirà el seu rumb fins a trobar la següent marca lateral corresponent a la balisa numero 2, de la mateixa manera que en el cas anterior, comprovarà que hi hagi un "1" a la posició dos del vector objectius:

0	<b>0</b>	1	0	0	0	0	9
---	----------	---	---	---	---	---	---

- Entrar a la rutina de "gestionar estació" i girar a la esquerra cap a la balisa.
- **Seguir recte deixant de llarg la balisa.**

En aquest cas, el robot decidirà seguir recte ja que l'objectiu corresponent a la balisa numero 2 no és troba en els seus objectius marcats. Per tant, el robot, actualitzarà la seva variable de numero de balises per a saber on es troba, i navegarà durant 3 segons ignorant els seus sensors de gir, per a continuar recte fins a trobar-se amb la unió corresponent a la balisa numero 2, a on tornarà a activar-se la rutina de "seguir recte" per gestionar la unió i recuperar la rutina de navegació bàsica.

En el moment en que es trobi la següent marca lateral, es repetirà la comprovació del vector objectius:

0	0	<b>1</b>	0	0	0	0	9
---	---	----------	---	---	---	---	---

- **Entrar a la rutina de "gestionar estació" i girar a la esquerra cap a la balisa.**
- Seguir recte deixant de llarg la balisa.

En aquest cas, com en el de la primera balisa, el robot haurà de girar per poder gestionar la balisa numero 3, repetint el mateix procés que en la balisa 1 i quedant el vector objectius de la següent manera:

0	0	<b>0</b>	0	0	0	0	9
---	---	----------	---	---	---	---	---

Una vegada gestionada la unió de final de balisa, el robot recupera el rumb de navegació fins trobar-se amb la marca lateral de la balisa 4. Com en el cas de la segona balisa no s'ha de gestionar aquesta, continuant així el rumb deixant l'estació 4 enrere.

0	0	0	<b>0</b>	0	0	0	9
---	---	---	----------	---	---	---	---

- Entrar a la rutina de "gestionar estació" i girar a la esquerra cap a la balisa.
- **Seguir recte deixant de llarg la balisa.**

Al finalitzar la unió de la balisa numero 4, el robot, seguirà el seu rumb, fins a trobar-se amb la parada de programació numero 2. Aquí el robot, tornarà a parar-se per a esperar possibles nous objectius. En aquest cas, per a poder provar la recirculació posterior s'afegeixen 2 nous objectius, teclejant per l'hyperterminal 2-4-9. Quedant el vector objectius de la següent manera:

0	<b>1</b>	0	<b>1</b>	0	0	0	9
---	----------	---	----------	---	---	---	---

Així doncs el robot recupera el seu rumb fins a arribar a la marca lateral de "parada de recirculació" a on s'agafaran els valors de les 7 primeres posicions del vector objectius i se sumaran per a poder decidir el següent:

<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	9
----------	----------	----------	----------	----------	----------	----------	---

Total = 2

- **Resultat de suma > 0 → Queden objectius pendents → Recirculació**
- Resultat de suma = 0 → Objectius aconseguits → Tornar a l'inici

Per tant, al arribar a la bifurcació de recirculació, el robot girarà per a seguir fins la següent balisa (numero 1) ja que li manquen objectius per a complir.

En aquest cas, es gestiona el gir a l'esquerra de la mateixa manera que en els casos anteriors, i s'actualitzarà la variable de control de balises per marcar-li al robot que la següent balisa que es trobi serà la numero 1.

Quan es trobi la marca d'unió de final de la recirculació, gestionarà la unió com si es tractes de la d'una balisa i recuperarà el camí inicial com si hagués sortit per primera vegada de la parada de programació numero 1.

Així doncs, el robot recuperarà el rumb fins a trobar-se amb la següent marca lateral, corresponent a la balisa numero 1, de la mateixa manera que anteriorment gestionarà les 4 balises de l'àrea de treball, entrant a la numero 2 i numero 4 per a complir els objectius que li falten i arribarà a la parada de programació numero 2 amb el vector objectius de la següent manera:

0	0	0	0	0	0	0	9
---	---	---	---	---	---	---	---

En aquest cas no afegirem nous objectius, per així donat per finalitzada la tasca del robot. Per tant es tecleja el numero 9 des de l'hyperterminal del ordinador, per marcar que no s'afegeixen objectius.

Llavors el robot, recuperarà el seu rumb fins arribar a la marca de recirculació, a on novament, es realitzarà la suma de les 7 posicions del vector objectius per comprovar si manquen objectius a completar:

0	0	0	0	0	0	0	9
---	---	---	---	---	---	---	---

Total = 0

- Resultat de suma > 0 → Queden objectius pendents → Recirculació
- **Resultat de suma = 0 → Objectius aconseguits → Tornar a l'inici**

Arribats en aquest punt, el robot no té objectius que complir, i per tant, no ha d'agafar la bifurcació de recirculació. Per tant s'activa la rutina de "seguir recte" per a deixar la bifurcació de recirculació enrere i es recupera el rumb de navegació basic tot actualitzant la variable de control de balises per a indicar-li al robot que la següent marca que es trobi, serà la parada de programació numero 1.

Així doncs el robot segueix el seu rumb fins a arribar a la següent marca lateral a on s'aturarà esperant nous objectius per a poder iniciar una altra vegada les tasques de compliment d'objectius.

Cal destacar que en qualsevol moment del flux d'execució esmentat anteriorment, el control d'obstacles està comprovant contínuament que no es trobi cap obstacle davant del robot. Si el robot es trobes un obstacle (en el cas de prova es va situar varies vegades i a diferents punts de l'àrea de treball una llibreta a mode de mur), aquest el detectaria i s'aturaria a una distància de 10 cm de l'obstacle, esperant a que aquest obstacle sigui retirat per mans externes i així continuar amb la rutina que estigues executant.



## 4.2 Errors durant el desenvolupament

Degut a la sensibilitat del hardware que s'ha fet servir, han aparegut errors de funcionament que feien que el desenvolupament no avances com teòricament s'esperava, ja que els servos encarregats de moure les rodes no funcionen de forma ideal, s'havia d'adaptar el programa de control de el que seria el nivell teòric, i introduir eines dins del software de control per reduir aquestes divergències.

D'altra banda un dels punts que ha donat més errors ha sigut la detecció de línies i de marques. Les parelles d'emissors/receptors d'infraroig son extremadament sensibles a la llum de l'entorn on es troben i condicionen la seva estabilitat de funcionament. Per tant ha calgut realitzar i calibrar aquets sensors sempre en el mateix espai i amb les mateixes condicions de llum, sinó s'havia de modificar el software per a cada entorn de llum a on es provés.

# Capítol 5. Conclusions

## 5.1 Compliment d'objectius

El fet de treballar mitjançant iteracions i fites de manera força constant ha donat flexibilitat a la metodologia de treball i ha facilitat el compliment dels objectius principals, ja que a mesura que es marcaven i s'assolien noves fites, els objectius es veien resolts així com la elaboració gradual de la memòria ha facilitat la tasca de redacció i no ha suposat quelcom farragós.

La majoria d'objectius i propostes que es van realitzar en les primeres reunions s'han vist realitzades composant el gruix principal de l'elaboració del projecte.

En el cas de que alguna proposta no s'hagi efectuat, era per manca de temps i recursos i ha passat a no ser descartada sino a passar al camp de possibles millores del projecte.

## 5.2 Desviacions

El pla de projecte no ha sofert grans desviacions en quan a hores planificades i en feina realitzada així com el flux de treball respecte de les reunions i les fites marcades.

Cal marcar el canvi de data inicial en quan a la entrega, ja que estava pensada per la convocatòria de juny, però per motius laborals no va ser possible realitzar-la en aquell període de temps, passant a la convocatòria de setembre.

En el cas de possibles desviacions en objectius, va haver un petit canvi en el que fa la comunicació entre robot i usuari, ja que inicialment estava previst realitzar aquesta comunicació mitjançant comandaments a distància, però per falta de recursos, es va haver de trobar una solució ràpida i eficaç que no va suposar cap canvi en l'elaboració del projecte i que a títol personal va suposar una millora en la implementació. Per tant es va substituir la comunicació per comandaments per la comunicació via bluetooth. Va suposar una desviació en especificacions tècniques eficaçment millorada per la proposta de part del director del projecte.

### 5.3 Línies d'ampliació

Qualsevol projecte d'aquest tipus, sempre és pot millorar d'alguna manera o altre. Ja sigui amb la millora dels components hardware aconseguint així millor rendiment, com realitzant millores conceptuals que afegeixen solidesa al projecte. Algunes d'aquestes millores conceptuals sonaven com a possibles objectius al iniciar el projecte, però malauradament, per la falta de temps i la planificació màxima en hores d'aquests tipus de projecte, fan que s'hagi de prioritzar alguns objectius que resulten mes principals vers d'altres que poden ser millores futures.

Part d'aquestes propostes que resulten millores son les següents:

- Múltiples robots: Per a realitzar una simulació més realista del que podria ser un laboratori d'anàlisi mèdica amb AGVs caldria disposar de múltiples AGVs, aquest fet comporta intrínsecament les següents línies de millora.
- Ampliar entorn de treball: Degut a que múltiples AGVs es trobarien en l'àrea de treball, caldria modificar aquesta per a que pogués suportar més d'un robot, com ara disposar de zones d'avançament, o de vies amb més d'un carril.
- Comunicació entre robots: Per tal de poder dur a terme la simulació amb múltiples robots una opció per a poder gestionar l'espai de l'entorn de treball seria que els robots es poguessin comunicar entre ells per així poder prioritzar-se en cas de compartir recursos com ara la via per on es desplacen.
- Gestió externa: Una altre manera de poder gestionar la simulació múltiple, sense dotar de comunicació als robots, seria mitjançant un software extern que controlés les prioritats i les polítiques d'eficiència, i que per tant coordines els robots dins de l'entorn de treball.

Per tant aquestes possibles millores es tradueixen en un increment de la complexitat del projecte, que es veu directament implicat en les hores del mateix. Passant de ser així de objectius a assolir a línies de millora.

# Referències

- [1] Ll. Ribas-Xirgo and A. Miró-Vicente, “Multi-Agent Model of a Sample Transport System for Modular IVD Laboratories”, unpublished, from 2010/2011 course on *Multi-Physical Agent Systems*, Univ. Autònoma de Barcelona (UAB), May 2011.
- [2] “Estadísticas de empleo”, Trovit, 30 Nov. 2011.  
[on-line a [http://empleo.trovit.es/index.php/cod.jobs\\_stats](http://empleo.trovit.es/index.php/cod.jobs_stats)]
- [3] “Parallax inc.”, Parallax, 30 Nov. 2011.  
[on-line a <http://www.parallax.com/>]
- [4] “Ferreteria y suministros”, Neoferr 30 Nov 2011.  
[on-line a <http://www.neoferr.com>]
- [5] “Robótica con el Boe-Bot”, Parallax 13 Feb 2012.  
[on-line a <http://www.parallax.com/Portals/0/Downloads/docs/prod/robo/RoboticaConElBoe-Bot-v3.0.pdf>]
- [6] “BASIC Stamp Software”, Parallax 13 Feb 2012.  
[on-line a <http://www.parallax.com/basicstampsoftware>]
- [7] “Ultrasonic Sensor”, Parallax 28 Feb 2012.  
[on-line a <http://www.parallax.com/Portals/0/Downloads/docs/prod/audiovis/Distance28015.pdf>]
- [8] “EmbeddedBlue™ 500 User Manual”, A7 Engineering 28 Febrer 2012.  
[on-line a <http://www.parallax.com/dl/docs/prod/comm/eb500UserManual.pdf>]

# Annexos

A continuació s'adjunten les subrutines per separat que han anat confeccionant els mòduls de software per a realitzar el programa de control final.

## Annex I : Detector obstacles

```
' obstacle.bs2
' {$STAMP BS2}
' {$PBASIC 2.5}

time VAR Word
counter VAR Word
servodreta VAR Word
servoesquerra VAR Word

Main:

DO
  GOSUB Obstacle
  GOSUB Moure
  PAUSE 10
LOOP
END

Obstacle:

PULSOUT 15, 5
PULSIN 15, 1, time
time = time ** 2251
DEBUG CR, "Distance = ", DEC4 time, " cm"      'transformació de distancia a cm
PAUSE 10

IF (time) < 20 THEN                             ' si menor de 10 cm atura
  servodreta = 755
  servoesquerra = 739
ELSE                                             ' tira recte
  servodreta = 650
  servoesquerra = 850
ENDIF
RETURN

Moure:

FOR counter = 1 TO 16                          'moure durant 1/4 de segon
  PULSOUT 13, servodreta
  PULSOUT 12, servoesquerra
  PAUSE 20
NEXT
RETURN
```

## Annex II: Navegació aleatòria

```

' Robotica con el Boe-Bot - MovementsWithSubrutinas.bs2|
' {$STAMP BS2}
' {$PBASIC 2.5}
DEBUG "Program Running!"
counter VAR Word

FREQOUT 4, 2000, 3000 ' señal de programa en inici/reset.

GOSUB Forward
GOSUB Left
GOSUB Right
GOSUB Backward
END

Forward:
FOR counter = 1 TO 64 ' 1 seg aprox
PULSOUT 13, 850
PULSOUT 12, 650
PAUSE 20
NEXT
PAUSE 200
RETURN
Left:
FOR counter = 1 TO 24
PULSOUT 13, 650
PULSOUT 12, 650
PAUSE 20
NEXT
PAUSE 200
RETURN
Right:
FOR counter = 1 TO 24
PULSOUT 13, 850
PULSOUT 12, 850
PAUSE 20
NEXT
PAUSE 200
RETURN
Backward:
FOR counter = 1 TO 64
PULSOUT 13, 650
PULSOUT 12, 850
PAUSE 20
NEXT
RETURN

```

## Annex III: Detector de marques

```

' -----[ Titulo ]-----
' {$STAMP BS2} ` Directiva Stamp.
' {$PBASIC 2.5} ` Directiva PBASIC.
' -----[ Variables ]-----
freqSelect VAR Nib
irFrequency VAR Word
irDetectLeft VAR Bit
irDetectRight VAR Bit
distanceLeft VAR Nib
distanceRight VAR Nib

irDetectCentre VAR Bit
distanceCentre VAR Nib

' -----[ Inicializacio ]-----
DEBUG CLS,
"IR objeto ZONE", CR,
"Left Centre Right", CR,
"-----"
' -----[ Rutina Principal ]-----
DO
GOSUB Get_Distances
GOSUB Display_Distances
LOOP

' -----[ Subrutina - Get_Distances ]-----
Get_Distances:
distanceLeft = 0
distanceRight = 0
distanceCentre = 0
FOR freqSelect = 0 TO 4
LOOKUP freqSelect, [37500,38250,39500,40500,41500], irFrequency
FREQOUT 9,1,irFrequency
irDetectLeft = IN10
distanceLeft = distanceLeft + irDetectLeft
FREQOUT 2,1,irFrequency
irDetectRight = IN3 ' amb l'esquema es el port0 pero el fara servir el BTH
distanceRight = distanceRight + irDetectRight

FREQOUT 7,1,irFrequency
irDetectCentre = IN8 ' amb l'esquema es el port0 pero el fara servir el BTH
distanceCentre = distanceCentre + irDetectCentre

PAUSE 100
NEXT
RETURN

' -----[ Subrutina - Display_Distances ]-----
Display_Distances:
DEBUG CR$RXY,2,3, DEC1 distanceLeft,
CR$RXY,9,3, DEC1 distanceCentre,
CR$RXY,14,3, DEC1 distanceRight
RETURN

```

## Annex IV: Seguidor de línies + detecció obstacles

```

' -----[ Titlulo ]-----
' {$STAMP BS2} ' Directiva Stamp.
' {$PBASIC 2.5} ' Directiva PBASIC.
DEBUG "Program Running!"
' -----[ Constantes ]-----
Kpl CON 35 ' Cambio de -35 to 35
Kpr CON -35 ' Cambio de 35 to -35
SetPoint CON 2 ' Cambio de 2 to 3.
CenterPulse CON 750

' -----[ Variables ]-----
freqSelect VAR Nib
irFrequency VAR Word
irDetectLeft VAR Bit
irDetectRight VAR Bit
distanceLeft VAR Nib
distanceRight VAR Nib
pulseLeft VAR Word
pulseRight VAR Word

irDetectCentre VAR Bit
distanceCentre VAR Nib

time VAR Word
counter VAR Word
' -----[ Inicializacion ]-----
FREQOUT 4, 2000, 3000
' -----[ Rutina Principal ]-----
DO
  GOSUB Get_Ir_Distances
  ' Calcula salida proporcional.
  pulseLeft = SetPoint - distanceLeft * Kpl + CenterPulse
  pulseRight = SetPoint - distanceCentre * Kpr + CenterPulse
  GOSUB Obstacle
  GOSUB Send_Pulse
LOOP
' -----[ Subrutina - Get IR Distances ]-----
Get_Ir_Distances:
  distanceLeft = 0
  distanceRight = 0
  distanceCentre = 0
  FOR freqSelect = 0 TO 4
    LOOKUP freqSelect, [37500,38250,39500,40500,41500], irFrequency
    FREQOUT 9,1,irFrequency
    irDetectLeft = IN10
    distanceLeft = distanceLeft + irDetectLeft
    FREQOUT 2,1,irFrequency
    irDetectRight = IN3 ' amb l'esquema tutorial es el port0 pero el fara servir el BTH
    distanceRight = distanceRight + irDetectRight
    FREQOUT 7,1,irFrequency
    irDetectCentre = IN8 ' amb l'esquema tutorial es el port0 pero el fara servir el BTH
    distanceCentre = distanceCentre + irDetectCentre
  PAUSE 5
  NEXT
RETURN

```



```

'-----[ Subrutina - Obstacle ]-----
Obstacle:
  PULSOUT 15, 5
  PULSIN 15, 1, time
  time = time ** 2251
  DEBUG CR, "Distance = ", DEC4 time, " cm"          'transformació de distancia a cm
  PAUSE 10
  IF (time) < 20 THEN                                ' si menor de 10 cm atura
    pulseRight = 755
    pulseLeft = 739
  ENDIF
  RETURN

' -----[ Subrutina - Get Pulse ]-----
Send_Pulse:
  PULSOUT 12,pulseLeft
  PULSOUT 13,pulseRight
  PAUSE 5
  RETURN

```

## Annex V : Comunicació Bluetooth

```

| { $STAMP BS2 }

'Wait for the eb500 radio to be ready
PAUSE 1000
'Connect to the remote device
SEROUT 1,84,["CON 74:F0:6D:E7:FC:3B",CR]
SERIN 0,84,[WAIT("ACK",CR)]
'Wait for the connection to be established and switch to data mode
WaitForConnection:
  IF IN5 = 0 THEN WaitForConnection
  HIGH 6
  PAUSE 300
  'Wait for 20 seconds
  PAUSE 20000
  'Switch to Command Mode
  LOW 6
  SERIN 0,84,[WAIT(CR,">")]
  'Disconnect from the remote device
  SEROUT 1,84,["dis",CR]
  SERIN 0,84,[WAIT(CR,">")]

```

Enric Marco Nin

Sabadell, setembre de 2012